

Fast approximation algorithms for p -centres in large δ -hyperbolic graphs

Katherine Edwards*, W. Sean Kennedy†, Iraj Saniee†

ke@princeton.edu, {kennedy,iis}@research.bell-labs.com

Abstract

We provide a quasilinear time algorithm for the p -center problem with an additive error less than or equal to 3 times the input graph's hyperbolic constant. Specifically, for the graph $G = (V, E)$ with n vertices, m edges and hyperbolic constant δ , we construct an algorithm for p -centers in time $O(p(\delta + 1)(n + m) \log(n))$ with radius not exceeding $r_p + \delta$ when $p \leq 2$ and $r_p + 3\delta$ when $p \geq 3$, where r_p are the optimal radii. Prior work identified p -centers with accuracy $r_p + \delta$ but with time complexity $O((n^3 \log n + n^2 m) \log(\text{diam}(G)))$ which is impractical for large graphs.

1 Introduction

The p -center algorithm is a discrete variant of arguably one of the most frequently used clustering algorithms, the k -means clustering. The goal of the p -center algorithm is to identify on a given graph a pre-specified number p of vertices or centers, such that the maximum distance of any graph vertex to its nearest p -center is minimized. For any given p , the algorithm naturally partitions a graph into p clusters induced by the position of its p -centers. Clusters induced by the p -centers are not necessarily balanced as these are determined strictly by the metric properties of the graph. Thus p -center clustering is more appropriate for distance-based partitioning or classification than other frameworks, such as community detection. Unfortunately, as a clustering algorithm the complexity of the p -center algorithm is generally prohibitive, $O(n^p)$ for an n -node graph, making it inapplicable to even moderate size graphs.

Proved nearly four decades ago, Shier's minimax result for trees and metric trees [16] leads to an exact algorithm with quasilinear time complexity (in the number of vertices and edges of the graph) for determination of an optimal set of p -centers by repeatedly finding diagonal pairs on the graph and carving out a ball containing one end of the current diagonal pair.

*Department of Computer Science, Princeton University, Princeton, NJ 08540

†Mathematics of Networks and Communications Department, Bell Labs, Nokia, NJ 07974

Hochbaum and Shmoys [11] give a (multiplicative) 2-approximation algorithm for determining p -centres in graphs satisfying the triangle inequality with running time $O(m \log_2 m)$. Subsequently, Dyer and Frieze [5] improve this to a 2-approximation algorithm with running time $O(np)$. These algorithms are, in a sense, best possible as Hsu and Nemhauser [12] show that determining an α -approximate solution to p -centers is NP-hard whenever $\alpha < 2$.

In an insightful paper [4], Chepoi and Estellon essentially apply the technique of Shier [16] to graphs with small hyperbolic constant, δ . These are graphs whose metric structure differs from the metric structure of a tree by a fixed constant (as explained in Section 2 and, in particular, Section 2.2 and Figure 1. For more details see [1, 4, 10]). The algorithmic version of this scheme [3] gives rise to what is essentially an $O(n^3)$ approximation for p -center on an n -vertex graph with hyperbolic constant δ appearing both as a prefactor in the complexity expression and also in the degree of approximation in terms of an additive constant to the radius of the optimal p -center partition. Of course the polynomial time complexity $O(n^3)$ is still impractical for graphs of hundreds of thousands to millions of nodes as would be even a quadratic complexity approximation.

Since there is evidence that real-life networks extracted from social media, co-authorship and collaboration, friendship and many other settings, have small hyperbolic constants [14], it would be desirable to know if the cubic complexity is tight or can be further reduced, at least by negotiating on the degree of the approximation. In this paper we show that that by giving up to 3δ in the (additive) approximation, one can achieve a quasilinear time p -center approximation. As such, this scheme is the first p -center approximation applicable to large graphs, particularly when p is relatively small, for example in the range $10 - 10^4$ and n is large, for example, $10^5 - 10^9$ vertices.

In the following sections we describe how the cubic complexity of [4] to quasilinear reduction is achieved without adding more than 3δ to the radius of the optimal p -center clusters. In Section 2 we outline necessary definitions, in particular, for geodesic metric spaces (Section 2.1) and hyperbolicity (Section 2.2). We then turn to a more formal discussion of p -centers, p -packings, and the dual problems which take center stage in our discussion (Section 3). In Section 3.1 we focus on algorithms for solving and approximating these problems on δ -hyperbolic graphs. The formal statements of our main results are also found in Section 3.1. Section 4 contains the proofs of the main results. We finish in Section 5 with experimental validation of our algorithms.

2 Definitions and notation

Let $G = (V, E)$ be an undirected graph, with V the set of vertices and E the set of edges. To each edge uv , we associate a line segment of length 1, so that we may refer to any point on uv at distance t from u and $1 - t$ from v ($0 \leq t \leq 1$). This (uncountably infinite) set of points of G is denoted $A(G)$. We will use the notation $n = |V(G)|$ and $m = |E(G)|$. In this paper, the distance $d(u, v)$ between any two points u and v in $A(G)$ is the length of a shortest path between them in G . When u and v are vertices, we write $[u, v]$ to refer to a shortest (also called *geodesic*) path. Note that shortest paths need not be unique. For a

geodesic path $P = [u, v]$ and $i \in [0, d(u, v)]$, the point $P[i]$ is the one at distance i from u on P .

2.1 Geodesic metric spaces and graphs

Let (X, d) be a metric space. If x, y are points in X , a *geodesic segment* $[x, y]$, when it exists, is a continuous curve parametrized by the line segment $[a, b]$ of length $d = d(x, y)$. That is, a map $\rho : [0, d] \rightarrow X$ with $\rho(0) = x, \rho(d) = y$ and $d(\rho(s), \rho(t)) = |s - t|$ for each $s, t \in [0, d]$. A metric space is *geodesic* if there exists a geodesic segment joining every pair of points. Note that geodesic segments need not be unique, e.g. a diagonal pair of points on a cycle.

Any graph as we have defined above can be viewed as a geodesic metric space $(A(G), d)$. Such a metric space is called *graphic* and it will be convenient in what follows to think of graphs in this way. In a graphic metric space, a geodesic $[x, y]$ is simply a shortest path from x to y regardless of x and y being in $V(G)$ or in $A(G)$.

Let $S \subseteq X$ be compact. The *diameter* $\text{diam}(S)$ of S is the maximum length of a geodesic between two vertices in S . For $u \in S$, $F_S(u)$ is the set of points in S whose distance from u is maximum. Two points $u, v \in S$ are *diametrical* if $d(u, v) = \text{diam}(S)$. They are *locally diametrical* if $u \in F_S(v)$ and $v \in F_S(u)$. It follows that $d(u, v \in F_S(u)) \leq \text{diam}(S)$ and $d(v, u \in F_S(v)) \leq \text{diam}(S)$.

If v is a point of $A(G)$ and $r \in \mathbb{R}$, we write $B_r(v)$ for the closed ball of radius r about v , i.e. all points at distance at most r from v . For a geodesic path $P = [u, v]$ and for the length $0 \leq \theta < d(u, v)$, the point $i = [u, v][\theta] \in A(G)$ is at distance θ from u on P . When there is no ambiguity, we identify the point $i = P[\theta]$ with the length θ . Clearly the two points $[u, v][i]$ and $[v, u][i]$ do not generally coincide.

2.2 Hyperbolicity

The concept of hyperbolicity of a metric space was introduced by Rips and Gromov in [10]. There are several essentially equivalent definitions but in this paper we will mainly use the *δ -thin-triangle* characterization.¹ For points x, y, z in X , we write $\Delta(x, y, z)$ to denote a *geodesic triangle* formed by x, y, z ; that is the union of three geodesics $[x, y], [y, z], [x, z]$ (usually the choice of geodesics won't matter).

Given a geodesic triangle $\Delta \equiv \Delta(x, y, z)$, let π be half the perimeter, $\pi = \frac{1}{2}(d(x, y) + d(y, z) + d(x, z))$ and define $\alpha_x = \pi - d(y, z)$ and similarly $\alpha_y = \pi - d(x, z)$ and $\alpha_z = \pi - d(x, y)$. Thus $\alpha_x + \alpha_y = d(x, y)$ and so on. One can imagine a triangle drawn in the Euclidean plane with side lengths $d(x, y), d(x, z)$ and $d(y, z)$. Its inscribed circle would touch the triangle sides $[x, y], [y, z]$ and $[z, x]$ at points m_z, m_x and m_y respectively. From elementary geometry, $[x, y][\alpha_x] = [y, x][\alpha_y] = m_z$ and $[y, z][\alpha_y] = [z, y][\alpha_z] = m_x$ and $[z, x][\alpha_z] = [x, z][\alpha_x] = m_y$, as illustrated in Figure 1.

The points m_x, m_y, m_z are called the *internal points* and $\alpha_x, \alpha_y, \alpha_z$ the *internal distances* corresponding to x, y, z respectively in Δ . The *insize* of the triangle Δ is the maximum of

¹For a comprehensive treatment of δ -hyperbolicity see [1].

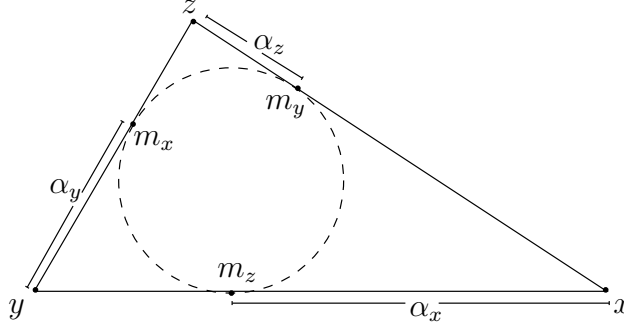


Figure 1: A geodesic triangle $\Delta(x, y, z)$ with internal points m_x, m_y and m_z and internal distances α_x, α_y and α_z labelled.

$$\max_{\theta \in [0, \alpha_x]} d([x, y][\theta], [x, z][\theta]), \max_{\theta \in [0, \alpha_y]} d([y, x][\theta], [y, z][\theta]), \text{ and } \max_{\theta \in [0, \alpha_z]} d([z, x][\theta], [z, y][\theta]).$$

Definition 1. Let (X, d) be a geodesic metric space, and $\delta \geq 0$. X is δ -hyperbolic (equivalently, the hyperbolicity of X is δ) if the insize of every geodesic triangle is at most δ . Let δ be minimum such that the insize of every geodesic triangle is at most δ . We say that X is δ -hyperbolic (equivalently, the hyperbolicity of X is δ).

If G is a graph whose associated graphic metric space is δ -hyperbolic then we say G is δ -hyperbolic. The reader may verify that every tree is 0-hyperbolic. Hyperbolicity is sometimes defined in terms of a four-point condition.

Lemma 2 (4-point condition, see Proposition 1.22 in [1]). Let (X, d) be a δ -hyperbolic metric space. There is a constant $\delta_{4\text{-point}} \leq \delta$ such that for any 4 points $x, y, z, w \in X$, their ordered set of sums of opposite sides, wlog $d(x, y) + d(w, z) \geq d(x, z) + d(y, w) \geq d(x, w) + d(y, z)$, satisfy $d(x, y) + d(w, z) - d(x, z) - d(y, w) \leq 2\delta_{4\text{-point}}$.

The fact that in a δ -hyperbolic metric space $\delta_{4\text{-point}}$ is always less than or equal to δ follows directly from the proof of Proposition 1.22 on page 411.

3 p -centers and p -packings

Let (X, d) be a geodesic metric space and S be a compact subset of X . Throughout this paper we rely on two intimately related notions, p -centers and p -packings.

Definition 3 (p -centers). A set $C \subset X$ r -dominates S if for every point $s \in S$ there exists a point $c \in C$ with $d(s, c) \leq r$. The p -radius of S , denoted by $r_p(S)$, is the minimum r such that there exists a set of at most p points $C_p(S)$ that r -dominates S . The points in $C_p(S)$ are called p -centers of S .

Definition 4 (*p-packings*). A set $D \subseteq S$ is an *r-dispersion* in S if each pair of points $s, s' \in D$, $s \neq s'$, $d(s, s') \geq r$. The *p-diameter* of S , denoted by $d_p(S)$, is the maximum r such that there exists a set of at least p points $D_p(S)$ that is an *r-dispersion* in S . The points in $D_p(S)$ are called a *p-packing*.

Consider a set of p points C which *r-dominate* S . By definition, for any choice of $p + 1$ points D , each $d \in D$ is within r of some $c \in C$, and by the pigeonhole principle, at least two, say a_1 and a_2 , are within r of the same $c \in C$. Hence,

$$d(a_1, a_2) \leq d(a_1, c) + d(a_2, c) \leq 2r.$$

So, $\min_{i \neq j} d(a_i, a_j) \leq 2r$. Since this holds for all choices of C and D , we have the following observation which first appeared in [16].

Observation 5. $r_p(S) \geq \frac{1}{2}d_{p+1}(S)$.

It turns out that these two invariants are equal whenever S has a tree-metric. Indeed, Shier showed the following.

Theorem 6 (Shier [16]). *Let T be a tree. Then $r_p(T) = \frac{1}{2}d_{p+1}(T)$.*

As discussed in Section 2.2, δ -hyperbolic spaces are treelike, by which we mean that they possess a metric structure that differs from a tree metric by δ . Therefore, it is logical to attempt to extend Shier's result on *p-center* covering and packing to such structures. Chepoi and Estellon [4] do exactly this by giving an elegant extension of Shier's theorem to δ -hyperbolic spaces.

Theorem 7 (Chepoi and Estellon [4]). *Let X be a δ -hyperbolic metric space and S a finite subset of X . Then*

$$r_p(S) \leq \frac{1}{2}d_{p+1}(S) + \delta$$

This relationship between $r_p(S)$ and $d_{p+1}(S)$ is a key element in algorithms for approximating *p-centers* and *p-packing*.

3.1 Algorithms for *p-centers* and *p-packings*

The *p-packing* problem, sometimes referred to as the *p-dispersion* problem, has received some attention in the literature. For example it is known to be NP-hard [6]. Highly relevant to our work is the heuristic that iteratively adds each of the p points by maximizing the points' distance from previously chosen points (see for example [7, 15]). This heuristic is shown to be a 2-approximation algorithm by Ravi, Rosenkrantz and Tayi [15]. For more information, we refer the interested reader to [8] that has an empirical comparison of ten *p-dispersion* heuristics.

To our knowledge, the previous best algorithm in terms of an additive error not exceeding δ for the *p-radius* follows from the Chepoi-Estellon bound (Theorem 7). Indeed, the proof in [4] leads to a polynomial algorithm to solve *p-centres* in graphs with an additive error

of δ on the p -radius.² Specifically, in time $O((n^3 \log n + n^2 m) \log(\text{diam}(G)))$ the authors in [4] determine a set U of p points such that U $(r_p + \delta)$ -dominates $V(G)$. Their algorithm involves finding diametrical pairs of vertices in subsets of $V(G)$ $O(n \log(\text{diam}(G)))$ times. Johnson's algorithm [13] finds the diameter in time $O(n^2 \log n + nm)$; hence the running time in Chepoi-Estillon [4] follows.

As pointed out in the introduction, in this work we leverage the fact that instead of finding diametrical pairs, one can just use **locally** diametrical pairs (introduced in Section 2.1) with significant reduction in computational time with only a small penalty in the p -radius. Our main result is the following.

Theorem 8. *Let G be a δ -hyperbolic graph, $p \geq 3$ an integer and $r_p(G)$ the optimal radius of the p -center for $V(G)$. There exists an algorithm to find a set of p points that $(r_p + 3\delta)$ -dominates $V(G)$. Further, the algorithm runs in time $O(n \log n + (m + n)((2p + 1)(\lceil 4 + 3\delta + 2\delta \log_2 n \rceil) + (p + 1))) = O(p(\delta + 1)(m + n) \log n)$.*

Though the Chepoi-Estillon algorithm [4] achieves a better approximation (an additive factor of δ instead of our 3δ), its running time is $O((n^3 \log n + n^2 m) \log(\text{diam}(G)))$. We first show below how to improve their running time by a factor of n (Lemma 11), but this approach still remains infeasible for large graphs. When $p \in \{1, 2\}$ we can achieve the same Chepoi-Estillon p -radius bound but in quasilinear time.

Theorem 9. *Let (X, d) be a δ -hyperbolic metric space, S a finite subset of X and $p \in \{1, 2\}$. There exists an algorithm to determine a set of p points that $(r_p + \delta)$ -dominate S . Further, the algorithm runs in time $O((2\delta + 1)t_X)$, where t_X is the time required to find the set of points at maximum distance from a given point in X . In particular in a δ -hyperbolic graph the running time is $O((2\delta + 1)(m + n))$.*

For $p = 1$, the previous best algorithm we know of is due to Chepoi et al. [3]: the approximation error is $\leq 5\delta$, and the computation requires just two breadth-first searches. In contrast, we require $2\delta + 1$ breadth-first searches to achieve the smaller additive factor of δ .

The remainder of this section is organized as follows. We start by showing how to improve the time complexity of the Chepoi-Estillon algorithm by only approximately finding diametrical pairs of vertices, that is via finding locally diametrical pairs. In the proofs of our main results, we will repeatedly apply this idea, showing that it is sufficient to solve the easier and computationally more efficient approximate version of this expensive sub-problem. We then move on to proofs of Theorems 9 and 8 in Sections 4 and 4.1, respectively.

Recall from Section 2.1 that a pair of vertices $\{u, v\}$ is *locally diametrical* if there is no vertex w such that $d(u, w) > d(u, v)$ or $d(v, w) > d(v, u)$. Clearly a diametrical pair is locally diametrical but the converse is not true in general (e.g., a cycle with handles). It turns out to be sufficient to find locally diametrical pairs in the main lemma of [4]. Indeed, the following

²The cited result also gives rise to an algorithm for general δ -hyperbolic spaces whose running time depends on the time to compute $F_S(x)$ for $x \in X$ and $S \subseteq X$. Because our interest is primarily in graphs, we direct the reader to [4] for details.

lemma is simply Lemma 1 from [4], but with the requirement that u and v be diametrical replaced with the weaker property of being locally diametrical.

Lemma 10. *Let X be a δ -hyperbolic metric space and $S \subseteq X$ be a compact set and $r \in \mathbb{R}$. Suppose that u and v are locally diametrical in S and let $[u, v]$ be a geodesic. Let $c = [u, v][r]$. Then $B_{2r}(u) \cap S \subseteq B_{r+\delta}(c) \cap S$.*

The proof of Lemma 1 in [4] works essentially unchanged to prove Lemma 10 by replacing diametrical pairs with locally diametrical pairs. Since we will use a refined version of the same argument that is needed for Lemma 10 in the proof of Theorem 8, we skip the proof of Lemma 10. We prove below (Lemma 12) that we can find a locally diametrical pair with at most $2\delta + 1$ breadth-first searches. Hence, we achieve the following significant reduction in the run time of the Chepoi-Estillon algorithm.

Lemma 11. *Let G be a δ -hyperbolic graph and p an integer. There exists an algorithm to find a set of p points that $(r_p + \delta)$ -dominates $V(G)$ that runs in time $O(n^2 \log(\text{diam}(G))(2\delta + 1))$.*

It remains to show how to efficiently determine locally diametrical pairs.

Lemma 12. *Given a δ -hyperbolic graph G and $S \subseteq V(G)$. There is an algorithm that finds a locally diametrical pair of vertices by performing at most $2\delta + 1$ breadth-first searches; that is, the running time is $O((2\delta + 1)(m + n))$.*

Proof. Choose a vertex $u \in S$ arbitrarily and find a vertex $v_1 \in F_S(u)$ by BFS. Then, find $v_2 \in F_S(v_1)$. Next, find a vertex $v_3 \in F_S(v_2)$. If $d(v_2, v_3) = d(v_1, v_2)$, then let $v = v_1$ and $w = v_2$ and we have found a locally diametrical pair. Otherwise $d(v_2, v_3) > d(v_1, v_2)$ and continue the process until v_k, v_{k+1} are found such that $d(v_k, v_{k+1}) = d(v_k, F_S(v_k))$ and $d(v_k, v_{k+1}) = d(v_{k+1}, F_S(v_{k+1}))$. This must happen for at most $k \leq \text{diam}(S)$. But by Proposition 3 in [3] $d(v_1, v_2) \geq \text{diam}(S) - 2\delta_{4\text{-point}} \geq \text{diam}(S) - 2\delta$ so k cannot exceed 2δ . This means no more than $(2\delta + 1)$ BFS steps or no more than $O(2\delta + 1)(m + n)$ steps are needed for finding a locally diametrical pair starting from $u \in S$. Then algorithm returns the locally diametrical pair (v_k, v_{k+1}) . \square

4 Approximating p -centers

In general, in searching for p -centers, first we approximately solve the dual problem, that is, we find D , a $(p + 1)$ -packing, with $|D| \geq p + 1$ such that

$$\{\max r \mid d(s, s') \geq r, \forall s \neq s' \in D\} \leq d_{p+1}(V).$$

This together with Observation 5 yields

$$\frac{1}{2} \{\max r \mid d(s, s') \geq r, \forall s \neq s' \in D\} \leq r_p(V). \quad (1)$$

Given these $(p+1)$ -points we find a set of p -points C such that setting $\lambda = \frac{1}{2} \{\max r \mid d(s, s') \geq r, \forall s \neq s' \in D\}$,

1. C λ -dominates the points in D , and
2. for each $a \in D$ there exists some $a' \in D$ and $c \in C$ such that c is on a geodesic between a and a' .

We prove later that these two properties together with δ -hyperbolicity allow us to show that for a carefully-selected set D , the p points in C $(\lambda + 3\delta)$ -dominate V , that is,

$$\{\min r \mid \text{for each } x \in V, \exists c \in C \text{ with } d(x, c) \leq r\} \leq \lambda + 3\delta. \quad (2)$$

Substituting the value of λ in (2) and applying (1) yields,

$$\begin{aligned} \{\min r \mid \text{for each } x \in V, \exists c \in C \text{ with } d(x, c) \leq r\} &\leq \frac{1}{2} \{\max r \mid d(s, s') \geq r, \forall s \neq s' \in D\} + 3\delta \\ &\leq r_p(V) + 3\delta. \end{aligned}$$

It follows that C $(r_p(V) + 3\delta)$ -dominates V as desired.

We now apply this approach to find a 1-center of a graph.

Theorem 13. *Let G be a δ -hyperbolic graph. There exists an algorithm to find a point c that $(r_1 + \delta)$ -dominates $V(G)$. The algorithm requires time $O((2\delta + 1)(m + n))$.*

Proof. Let x, y be a locally diametrical pair of vertices and let $[x, y]$ be a geodesic segment. As described above, set $\lambda = \frac{d(x, y)}{2}$ and choose $c = [x, y][\lambda]$. Clearly, $C = \{c\}$ satisfies Properties 1 and 2 above. We now show that $C = \{c\}$ $(\lambda + \delta)$ -dominates V .

Let z be any point in V and consider the geodesic triangle $\Delta(x, y, z)$ as depicted and labeled in Figure 2. Without loss of generality, assume that $d(y, z) \leq d(x, z)$. Since (x, y) is locally diametrical, then

$$d(y, z) \leq d(x, z) \leq d(x, y)$$

which implies that

$$\alpha_z \leq \alpha_y \leq \alpha_x.$$

(This means that in the figure c lies to the right of m_z , as shown.) Then

$$d(z, c) \leq \alpha_z + \delta + d(c, m_z) \leq \alpha_z + \delta + \lambda - \alpha_y \leq \delta + \lambda.$$

As the claim holds for any z , c $(\lambda + \delta)$ -dominates $V(G)$, and therefore, since $\lambda = \frac{1}{2}d(x, y) \leq \frac{1}{2}d_2(V) \leq r_1(V)$, the latter inequality by Observation 5, and thus c $(r_1 + \delta)$ -dominates $V(G)$, as desired. To complete the proof, we note that by Lemma 12, x and y can be found in time $O((2\delta + 1)(m + n))$. \square

We note that in the course of the above prove we demonstrated the following fact that we shall reuse.

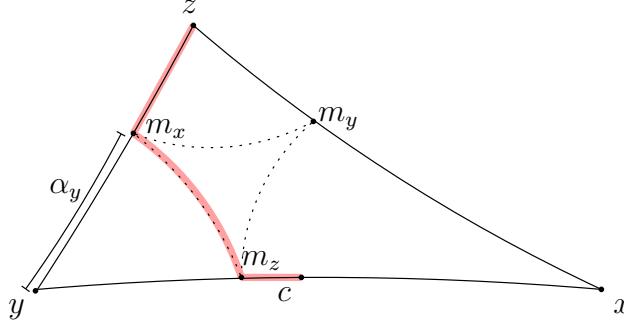


Figure 2: A geodesic triangle $\Delta(x, y, z)$ with internal points m_x, m_y, m_z and c labelled as in the proof of Theorem 13. Dashed lines indicate a distance $\leq \delta$ and the red line indicates the upper estimate for $d(z, c)$.

Observation 14. *Let z be any vertex in $V(G)$, (x, y) a locally diametrical pair of vertices, $c \in A(G)$ the mid-point of $[x, y]$ and $\lambda = \frac{d(x, y)}{2}$. Then $d(z, c) \leq \lambda + \delta$.*

In extending these proof techniques to the general case for $p > 1$, we run into the following two difficulties, each costing us an additional δ in our approximation error. First, Property 2 only guarantees that p of the $\binom{p+1}{2}$ pairs of points in D have a geodesics connecting them containing some point $c_i \in C$. This will force us use two geodesic triangles to bound the distance from some points in V to their closest center in C . Second, in achieving the quasilinear runtime, we are only able to find a $(\lambda + 2\delta)$ -approximation for the $(p + 1)$ -packing problem. We omit further details until Section 4.1.

To finish off this section, we prove that when $p = 2$ we can find a 2-center solution which $(r_2 + \delta)$ -dominates G . Like Theorem 13, this is stronger than our general result (Theorem 8) and the proof does not use the machinery outlined at the beginning of Section 4 that relies on Properties 1 and 2. Theorems 13 and 15 may be special cases of a general and stronger result than our main result, so we include it.

Theorem 15. *Let G be a δ -hyperbolic graph. There exists an algorithm to find points c_1, c_2 that $(r_2 + \delta)$ -dominate $V(G)$. The algorithm requires time $O((2\delta + 1)(m + n))$.*

Proof. Let x, y be a locally diametrical pair of vertices and let $[x, y]$ be a geodesic segment. Choose z so that $\min\{d(z, x), d(z, y)\}$ is maximized (requires two BFS). We let our 3-packing be $D = \{x, y, z\}$. Assume without loss of generality that $d(x, y) \geq d(x, z) \geq d(y, z)$, and so, $\lambda = \frac{1}{2}\{\max r \mid d(s, s') > r, \forall s \neq s' \in D\} = \frac{1}{2}d(y, z)$.

We choose $c_1 = [x, y][\lambda]$ and $c_2 = [y, x][\lambda]$. We claim that $C = \{c_1, c_2\}$ satisfy Equation 2, with $t = 1$, and so, C $(r_2 + \delta)$ -dominates G .

To prove the claim, let $\Delta_1 = \Delta(x, y, z)$ be a geodesic triangle. Let w be any point of G and let $\Delta_2 = \Delta(x, y, w)$ be a geodesic triangle so that Δ_1 and Δ_2 share the geodesic $[x, y]$. We will show that $\min\{d(w, c_1), d(w, c_2)\} \leq \lambda + \delta$. Take $\alpha_x, \alpha_y, \alpha_w$ and m_x, m_y, m_w to denote the internal distances and points in Δ_2 . Without loss of generality assume $d(w, x) \leq d(w, y)$

which implies that $d(w, x) \leq d(y, z) = 2\lambda$ and $\alpha_x \leq \alpha_y$. We distinguish two cases, as illustrated in Figure 3.

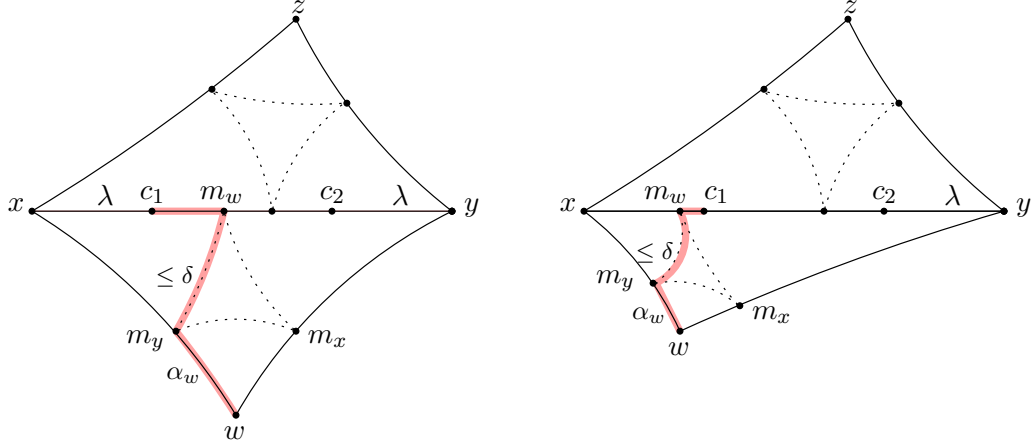


Figure 3: Figure for Cases 1 and 2 in the proof of Theorem 15. The red lines indicate the upper estimate for $d(w, c_1)$. Dashed lines indicate a distance $\leq \delta$.

Case 1: $\lambda < \alpha_x < d(x, y) - \lambda$

From the choice of z , it follows that either $d(w, x) \leq d(y, z) = 2\lambda$ or $d(w, y) \leq 2\lambda$. Assume without loss of generality that $d(w, x) = d(w, m_y) + d(m_w, x) \leq 2\lambda$. Therefore, $d(w, c_1) \leq d(w, m_y) + d(m_y, m_w) + d(m_w, c_1) \leq d(w, m_y) + \delta + d(m_w, x) - \lambda \leq \lambda + \delta$.

Case 2: $\alpha_x \leq \lambda$

In this case m_w lies between x and c_1 on the geodesic segment $[x, y]$. By the local maximality of x and y , we have $d(y, w) = \alpha_y + \alpha_w \leq \alpha_y + \alpha_x = d(x, y)$ and so $d(w, m_y) = \alpha_w \leq \alpha_x = d(x, m_w)$. Then $d(w, c_1) \leq d(w, m_y) + d(m_y, m_w) + d(m_w, c_1) \leq d(x, c_1) + \delta = \lambda + \delta$.

To complete the proof, we need only show that c_1, c_2 can be found in $O((2\delta + 1)(m + n))$ time. By Lemma 12, x and y can be found in time $O((2\delta + 1)(m + n))$ and the vertex z can be found by doing a breadth-first search rooted at x and one rooted at y . Given $D = \{x, y, z\}$, the vertices c_1 and c_2 can then be found by storing the last breadth-first search used in finding x and y and $\lambda = \frac{1}{2} \min\{d(x, x), d(y, z)\}$. The runtime now follows. \square

4.1 The general algorithm

Our algorithm and proof follow the same three basic steps, though each step is more involved. As a reminder these three steps are 1) approximately solving the dual problem, or finding a $(p + 1)$ -packing, 2) deriving p -points from this dual solution that satisfy Properties 1 and 2, and 3) bounding the approximation guarantee by showing Equation 2.

It turns out the difficult part of these three steps is Step 1. For this step, we need to extend the notion of a ‘locally diametrical pair’ to a ‘locally diametrical set’ in such a way

that i) it provides us with both the tools we need to satisfy Properties 1 and 2 and ii) it can be determined efficiently. We find a set of $(p + 1)$ vertices $D = \{v_0, v_1, \dots, v_p\}$ with

$$\lambda(D) := \frac{1}{2} \{ \max r \mid d(s, s') \geq r, \forall v_i \neq v_j \in D \}$$

such that the following three properties hold

- (a) (Vertex relabeling) $d(v_0, v_i) = 2\lambda(D)$ for some $v_i \in D$,
- (b) (Extending locally diametrical pairs to *locally diametrical sets*) For each $v_i \in D$ with $d(v_i, v_j) = 2\lambda(D)$ for some v_j , there exists no $w \in V(G)$ with $d(w, v_k) > 2\lambda(D)$, $\forall v_k \in D \setminus \{v_i\}$, and
- (c) (δ -hyperbolic version of locally diametrical sets) for each $i \geq 1$, there exists no vertex $v \in V(G)$ with $d(v_0, v) > d(v_0, v_i) + 2\delta$ and $d(v_i, v) \leq 2\lambda(D)$ and $d(v, v_j) > 2\lambda(D)$ for each $j \neq i$.

These three requirements provide us with what is needed to determine a set of $(p + 1)$ vertices satisfying Properties 1 and 2. Specifically, we prove

Lemma 16. *Let G be a δ -hyperbolic graph and $\Lambda_n = \lceil 4 + 3\delta + 2\delta \log_2 n \rceil$. There exists an algorithm to find a set D of $p + 1$ vertices satisfying (a), (b) and (c). The algorithm runs in time $O(n \log n + (m + n)((2p + 1)\Lambda_n + (p + 1)))$.*

Given a set of $p + 1$ vertices satisfying Properties (a), (b) and (c) it is straightforward to find $C = \{c_1, \dots, c_p\}$ satisfying Properties 1 and 2. For each $1 \leq i \leq p$, let c_i be the vertex at distance λ from v_i on the shortest path from v_i to v_0 , i.e. $c_i = [v_i, v_0][\lambda]$.

Lemma 17. *Let G be a δ -hyperbolic graph. Suppose that $D = \{v_0, v_1, \dots, v_p\}$ satisfy (a), (b) and (c). Then the set of p points $C = \{c_i \mid c_i = [v_i, v_0][\lambda]\}$ $(\lambda + 3\delta)$ -dominate G .*

As described above (beginning of Section 4), such C $(r_p(V) + 3\delta)$ -dominates V as desired. So, given the Lemmas 16 and 17, the proof of Theorem 8 follows once establishing the runtime, which we do now. First, determining the set D takes $O(n \log n + (m + n)((2p + 1)\Lambda_n + (p + 1)))$. Given D , the set of vertices $\{c_i, 1 \leq i \leq p\}$ can clearly be constructed by performing a breadth-first search rooted at v_0 . Theorem 8 now follows.

In the next two sections we establish Lemmas 16 and 17. Lemma 16 is the more interesting of the two proofs, and takes us deeper into the analysis of locally diametrical sets. The proof of Lemma 17 is a sophistication of the ideas in Theorems 13 and 15. We begin with that lemma.

4.2 Proof of Lemma 17

We show that every vertex of G is at distance at most $\lambda + 3\delta$ from some centre c_i . Let $w \in V(G)$ and suppose that w is at distance greater than $\lambda + 3\delta$ from each centre. Property (b) implies $d(w, v_i) \leq 2\lambda$ for some i . We prove below the following claim.

Claim 18. $d(w, v_j) > 2\lambda$ for each $j \neq i$.

Using the claim, we can prove Lemma 17. Consider the geodesic triangle $\Delta(v_i, v_0, w)$, and recall that c_i belongs to the geodesic $[v_i, v_0]$. There are two cases to handle.

First, suppose that $d(v_i, m_w) \geq \lambda$. Then a w - c_i -path can be constructed by concatenating the geodesics from $[w, m_{v_0}]$, $[m_{v_0}, m_w]$ and $[m_w, c_i]$, and so, since $d(w, v_i) \leq 2\lambda$

$$\begin{aligned} d(w, c_i) &\leq d(w, m_{v_0}) + d(m_{v_0}, m_w) + d(m_w, c_i) \\ &\leq d(w, m_{v_0}) + \delta + d(m_w, v_i) - \lambda \\ &\leq \lambda + \delta, \end{aligned}$$

a contradiction.

Otherwise, if $d(v_i, m_w) < \lambda$, then

$$\begin{aligned} \lambda + 3\delta < d(w, c_i) &\leq d(w, m_{v_0}) + d(m_{v_0}, m_w) + d(m_w, c_i) \\ &\leq d(w, m_{v_0}) + \delta + d(m_w, c_i). \end{aligned}$$

Since $d(v_i, c_i) = d(v_i, m_w) + d(m_w, c_i) = \lambda$, we deduce that $d(w, m_{v_0}) > d(v_i, m_w) + 2\delta$. It follows that $d(v_0, w) > d(v_0, v_i) + 2\delta$, which along with Claim 18, contradicts Property (c).

It follows that w is within $\lambda + 3\delta$ from at least one centre. We need only prove the claim.

Proof of Claim 18. Suppose that w is at distance at most 2λ from both v_i and v_j . Let c'_i and c'_j be the vertices at distance λ from i and j respectively on the geodesic $[v_i, v_j]$. We will show that at least one of $d(c_i, c'_i)$ and $d(c_j, c'_j)$ is at most δ . Consider the geodesic triangle $\Delta(v_i, v_j, v_0) = [v_i, v_j] \cup [v_i, v_0] \cup [v_j, v_0]$ and let $m_{v_i}, m_{v_j}, m_{v_0}$ be as described above. Assume for contradiction that both $d(c_i, c'_i)$ and $d(c_j, c'_j)$ are greater than δ . It follows that $d(v_i, m_{v_j}) < \lambda$ and $d(v_j, m_{v_i}) < \lambda$. But then $d(v_i, v_j) = d(v_i, m_{v_0}) + d(m_{v_0}, v_j) = d(v_i, m_{v_j}) + d(v_j, m_{v_i}) < 2\lambda$, a contradiction. Assume then, without loss of generality, that $d(c_i, c'_i) \leq \delta$.

Now consider the geodesic triangle $\Delta(v_i, v_j, w)$ and let m_w be defined as usual. First, suppose that $d(v_i, m_w) \geq d(v_i, c'_i)$. Then

$$d(w, c'_i) \leq d(w, m_{v_j}) + \delta + d(v_i, m_w) - d(v_i, c'_i) \leq d(w, v_i) + \delta - \lambda \leq \lambda + \delta.$$

Now, suppose that $d(v_i, m_w) < d(v_i, c'_i)$. Then

$$d(w, c'_i) \leq d(w, m_{v_i}) + \delta + d(v_j, m_w) - d(v_j, c'_i) \leq d(w, v_j) + \delta - \lambda \leq \lambda + \delta.$$

In either case, $d(w, c'_i) \leq \lambda + \delta$, and so $d(w, c_i) \leq \lambda + 2\delta$, a contradiction. \square

4.3 Proof of Lemma 16

A proof sketch is as follows. We first show that we can find a $(p+1)$ -packing that is within $O(\delta \log_2 n)$ of an optimal solution. To do so, we find a tree T which approximately preserves distances on our input graph G . It turns out that exactly solving the $(p+1)$ -packings on trees can be done efficiently, though in contrast to before, we solve the p -centres first and use

this to construct a dual solution in G . The fact that T is a good approximating tree allows us to bound how close our $(p+1)$ -packing is to an optimal solution and in turn helps us achieve the quasilinear running time. Finally, given this initial $(p+1)$ -packing, we iteratively improve the solution whenever possible until we achieve Properties (a), (b), (c). Clearly, (a) can hold for all solutions after relabelling, so the only difficulty is in insuring both (b) and (c) hold.

We will use the following theorem, which we will deduce from known results at the end of this section, to find our initial $(p+1)$ -packing. Let $\Lambda_n = \lceil 4 + 3\delta + 2\delta \log_2 n \rceil$.

Theorem 19. *There exists an algorithm to find a set \mathcal{P} of $p+1$ vertices satisfying $d(u, v) \geq \kappa, \forall u \neq v \in \mathcal{P}$, for some κ with $d_{p+1}(G) - \kappa \leq \Lambda_n$. The algorithm runs in time $O(n \log n)$.*

Given the set \mathcal{P} of $(p+1)$ -points from Theorem 19, we now describe an efficient iterative algorithm which finds $(p+1)$ -points satisfying Properties (a), (b) and (c). Our argument bounds the number of iterations using the following potential function.

Definition 20. *Let G be a graph and let $\mathcal{P} \subseteq V$ be a set of p vertices and suppose that κ is the largest value such that $d(u, v) \geq \kappa$ for all $u \neq v \in \mathcal{P}$. Let $\eta(\mathcal{P})$ denote the number of vertices in \mathcal{P} which are exactly at distance κ from at least one other vertex in \mathcal{P} . We define the potential of \mathcal{P} as $\phi(\mathcal{P}) := p(\kappa + 1) - \eta(\mathcal{P})$.*

Algorithm 1: Finding an initial set \mathcal{P} of vertices

Input: Graph $G = (V, E)$ and integer p .

Output: A set \mathcal{P} of $p+1$ vertices satisfying $d(u, v) \geq \kappa, \forall u \neq v \in \mathcal{P}$, for some κ with $d_{p+1}(G) - \kappa \leq \Lambda_n$.

Let $T = (V, F)$ be the tree determined by Theorem 21.

Let λ be the p -radius of the p -centers of T determined by Theorem 22.

Let \mathcal{P} be a set of maximum size s.t. $d(u, v) \geq 2\lambda$ for each $u \neq v \in \mathcal{P}$ (Theorem 23).

Let \mathcal{P}' be a set of $p+1$ unique vertices chosen arbitrarily from \mathcal{P} .

return $\mathcal{P} = \mathcal{P}'$

Algorithm 4 together with Subroutines 2 and 3 describe the algorithm. We first prove that if Algorithm 4 terminates then it is correct, that is, \mathcal{P}' satisfies (a), (b) and (c). The algorithm terminates if the potential $\phi(\mathcal{P})$ has not increased after successive executions of Subroutines 3 and 2. As Subroutine 2 executes last, the returned \mathcal{P} satisfies (a) and (b) as satisfying (b) is the stopping condition and, as mentioned above, (a) always holds after a relabelling. Since $\phi(\mathcal{P})$ is unchanged by Subroutine 2, \mathcal{P} is unchanged as well. For the purpose of analysis, we will adopt the following notation. Let $\{v_0, \dots, v_p\}$ be the labelling specified in the description of Subroutine 3. Then, for each $v_i (i \geq 1)$, if it was improved, let v'_i be the vertex v_i was replaced by. Otherwise write $v'_i = v_i$. So, now consider $\mathcal{P} = \{v_0, v'_1, \dots, v'_p\}$, that is output by Subroutine 3. For contradiction, suppose that \mathcal{P} does not satisfy Property (c). Then, there exist an index i and a vertex v''_i with $d(v_0, v''_i) > d(v_0, v'_i) + 2\delta$ and $d(v'_i, v''_i) \leq 2\lambda(\mathcal{P})$ and $d(v''_i, v'_j) \geq 2\lambda(\mathcal{P}), \forall j \neq i$. Further, by the choice of v'_i , there must exist some index

Subroutine 2: Satisfying Properties (a) and (b).

Input: A set \mathcal{P} satisfying Property (a) for $\lambda(\mathcal{P})$.

Output: A set \mathcal{P}' satisfying Property (a) and (b) for $\lambda(\mathcal{P}') \geq \lambda(\mathcal{P})$.

We say that a vertex $u \in \mathcal{P}$ is *improvable* to $w \notin \mathcal{P}$ if there exists $v \in \mathcal{P}$ with $d(u, v) = 2\lambda(\mathcal{P})$ and $d(w, x) > 2\lambda(\mathcal{P}), \forall x \in \mathcal{P} \setminus \{u\}$.

repeat

for i **from** 0 **to** p **do**

if v_i *is improvable to some* v **then**

 replace v_i in \mathcal{P} with the improved vertex ($\mathcal{P} = (\mathcal{P} \setminus v_i) \cup \{v\}$).

else

 do nothing.

end

until *no vertex is found to be improvable.*

return $\mathcal{P}' = \mathcal{P}$.

Subroutine 3: Satisfying Properties (a), (b) and (c).

Input: A set \mathcal{P} satisfying Property (a) and (b) for $\lambda(\mathcal{P})$.

For this step we label the vertices of \mathcal{P} in a specific way. Let v_0 and v_p be vertices in \mathcal{P} with $d(v_0, v_p) = 2\lambda(\mathcal{P})$. Then label the remaining vertices of \mathcal{P} as $\{v_0, v_1, \dots, v_p\}$ so that $d(v_0, v_i) \geq d(v_0, v_j)$ for each $i > j$.

In this context, we say that a vertex v_i ($1 \leq i \leq p$) is improvable to $v'_i \notin \mathcal{P}$ if $d(v_0, v'_i) > d(v_0, v_i)$ and $d(v_i, v'_i) \leq 2\lambda(\mathcal{P})$ and $d(v'_i, v_j) > 2\lambda(\mathcal{P})$ for each $j \neq i$.

for i **from** 1 **to** p **do**

if v_i *is improvable* **then**

 replace v_i in \mathcal{P} with the vertex v'_i furthest from v_0 that satisfies $d(v_i, v'_i) \leq 2\lambda(\mathcal{P})$ and $d(v'_i, v_j) > 2\lambda(\mathcal{P})$ for each $j \neq i$.

end

return \mathcal{P} .

Algorithm 4: Finding an optimal and optimized set of vertices

Input: A set \mathcal{P} satisfying Property (a) for $\lambda(\mathcal{P})$.

Output: A set \mathcal{P}' satisfying Property (a), (b) and (c) for $\lambda(\mathcal{P}') \geq \lambda(\mathcal{P})$.

Let \mathcal{P} **be** the returned set of Subroutine 2 with input \mathcal{P} .

repeat

Let \mathcal{P}' **be** the returned set of Subroutine 3 with input \mathcal{P} .

Let \mathcal{P} **be** the returned set of Subroutine 2 with input \mathcal{P}' .

until $\phi(\mathcal{P}') = \phi(\mathcal{P})$

return $\mathcal{P}' = \mathcal{P}$

$j > i$ with $d(v''_i, v_j) < 2\lambda(\mathcal{P})$. We will apply Lemma 2 to reach a contradiction, using the vertices v_0, v'_i, v''_i, v_j , as illustrated in Figure 4. By the choice of labelling, we have $2\lambda(\mathcal{P}) \leq d(v_0, v_j) \leq d(v_0, v_i) \leq d(v_0, v'_i) < d(v_0, v''_i) - 2\delta$. There are three distance sums to consider. We claim that $d(v_0, v'_i) + d(v'_i, v_j) > \max\{d(v_0, v'_i) + d(v''_i, v_j), d(v_0, v_j) + d(v'_i, v''_i)\} + 2\delta$. This is clear because both $d(v''_i, v_j) \leq 2\lambda(\mathcal{P})$ and $d(v'_i, v''_i) \leq 2\lambda(\mathcal{P})$ while $d(v'_i, v_j) \geq 2\lambda(\mathcal{P})$. By Lemma 2, this contradicts the δ -hyperbolicity of G . It follows that \mathcal{P} also satisfies (c).

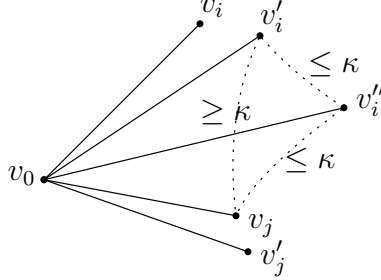


Figure 4: Proof that \mathcal{P} satisfies Property (c)

It remains to prove that the algorithm terminates and to bound the runtime. To see that Algorithm 4 terminates, we first note that whenever a vertex in \mathcal{P} is improved in Subroutine 2, the distance to its closest neighbour strictly increases. Therefore, after at most $p + 1$ rounds of the repeat until loop Subroutine 2, $\lambda(\mathcal{P})$ strictly increases. Further, each round (except the last one) in which the potential doesn't change is preceded by an iteration of Subroutine 3. By Theorem 19, for the initial $(p + 1)$ -points \mathcal{P}^* , $d_{p+1}(G) - 2\lambda(\mathcal{P}^*) \leq \Lambda_n$. Hence, $\phi(\mathcal{P}^*) \geq (p + 1)(2\lambda(\mathcal{P}^*) + 1) - (p + 1) = (p + 1)2\lambda(\mathcal{P}^*) \geq (p + 1)(d_{p+1}(G) - \Lambda_n)$. Further, any set of $p + 1$ vertices has dispersion at most $d_{p+1}(G)$ and therefore has potential at most $(p + 1)(d_{p+1}(G) + 1)$. We conclude that the repeat until loop of Algorithm 4 can be executed at most $(p + 1)\Lambda_n$ rounds in total.

We now examine the complexity of the algorithm. To obtain the initial set \mathcal{P} as in Theorem 19 takes time $O(n \log n)$. Given a set \mathcal{P} , we can determine and record the set of distances $\{d(v, v_i) : v \in V(G), 0 \leq i \leq p\}$ by performing a breadth-first search rooted at each vertex $v_i \in \mathcal{P}$. From these distances, it can easily be checked in linear ($O(n)$) time whether a vertex is improvable. To complete the first round the first time we perform Subroutine 2, we must perform $p + 1$ breadth-first searches. Each time a vertex is improved (in either Subroutine 2 or Subroutine 3), we need an additional one. From the discussion above it follows that at most $p + 1 + ((p + 1) + p)\Lambda_n$ breadth-first searches need be done. The algorithm therefore runs in time $O(n \log n + (m + n)((2p + 1)\Lambda_n + (p + 1)))$.

We now deduce Theorem 19 and its corresponding Algorithm 1. In finding our initial $(p + 1)$ -packing, we use the following definitions and results. For a graph G and constant k , we say that a tree T with vertex set $V(G)$ is a k -approximating tree if $|d_G(u, v) - d_T(u, v)| \leq k$ for every pair of vertices $u, v \in V$. Chepoi et al. showed in [3] that δ -hyperbolic graphs have good approximating trees that can be computed in linear ($O(m)$) time.

Theorem 21 ([3]). *Let $G = (V, E)$ be a δ -hyperbolic graph, and let $\Lambda_n = \lceil 4 + 3\delta + 2\delta \log_2 n \rceil$. There exists a Λ_n -approximating tree $T = (V, F)$ of G . Furthermore T can be computed from G in time $O(m)$.*

Fredrickson [9] showed that p -centres can be solved in linear time on trees.

Theorem 22 ([9]). *Let T be a tree and p an integer. There exists an algorithm to solve p -centres exactly on T in time $O(n)$.*

Shier proved in [16] (see Theorem 6) that in trees, the p -radius is always half the $p + 1$ -diameter. In [2] Chandrasekaran and Daughety gave an algorithm to find the p -diameter (and an optimal packing of size p) in a tree in time $O(n^2 \log n)$. Their technique involves a binary search for λ_p through repeated application of a subroutine which, when given a half-integer λ , produces a maximum number of points which are pairwise at distance $> 2\lambda$. More precisely,

Theorem 23 ([2]). *Let $T = (V, E)$ be a tree and let 2λ be an integer. There exists an algorithm which, in time $O(n \log n)$, produces a set $W \subseteq V$ of maximum size such that $d(u, v) \geq 2\lambda$ for each $u \neq v \in V$.*

Combining Theorem 23 with Theorem 6 and Theorem 22 we obtain an $O(n \log n)$ algorithm to find an optimal packing of size p in a tree.

Now, suppose that G is δ -hyperbolic and T is a Λ_n -approximating tree for G . By definition of an approximating tree, for every $u, v \in V$ we have $|d_T(u, v) - d_G(u, v)| \leq \Lambda_n$. It follows that $|d_{p+1}(T) - d_{p+1}(G)| \leq \Lambda_n$. Thus we obtain Algorithm 1 and Theorem 19 that yields our initial $(p + 1)$ -packing.

At this point, the reader may be wondering why we go to the trouble of Algorithm 1 to obtain our initial $(p + 1)$ -packing. Indeed, one could start with any packing at the beginning of Algorithm 4, and repeat rounds of Subroutines 2 and 3 until a packing satisfying Properties (a), (b) and (c) is found. However, as we have just seen, the number of times we may need to repeat the rounds is upper bounded by the difference between the dispersion of our initial set and the optimal dispersion d_{p+1} . When the initial set is chosen using Algorithm 1, this difference is $O(\delta \log n)$, whereas trying to save time choosing the initial set (say, by choosing it arbitrarily) may result in an additional linear factor in the complexity bound. Applying the greedy 2-approximation algorithm of Ravi, Rosenkrantz and Tayi mentioned in Section 3.1 adds a factor of d_{p+1} , which may also be linear. However, the practitioner may wish to experiment.

5 Empirical results

We have implemented the algorithms from Theorem 8 ($p \geq 3$) and Theorem 9 ($p \leq 2$). For comparison, we have also implemented the algorithms of Chepoi et al. (Ch.) [3] ($p = 1$) and Chepoi-Estellon (C-E) [4] ($p \geq 2$). We also compared Theorems 8 and 9 to the following

Table 1: Networks analyzed

Network	Type	$ V $	$ E $	diameter	radius	$\delta_{4-point}$
sprintlink-1239	Rocketfuel ISP network	8341	14025	13	7	3
p2p-gnutella25	peer-to-peer network	22663	54693	11	7	3
sn-medium	social network	26567	226566	14	7	4
web-stanford	web network	255265	1941926	164	82	1.5 (est.)

simple algorithm: Compute a distance approximating tree as in Theorem 21 and return an exact solution to p -centres on T .

We ran the algorithms on four graphs extracted from real networks arising from different types of data. All graphs are simple and have unit edge lengths and each has a small hyperbolicity constant. Table 1 briefly summarizes the networks we analyzed; more information about the data can be found in [14].³ In the case of the web-stanford graph, we have only an estimate of $\delta_{4-point}$ obtained by sampling since the graph is quite large. Table 2 contains a comparison of the estimated p -radius r_p of the three algorithms. We have run only our algorithm from Theorem 4.1 on the largest network (web-stanford), since the running time of C-E is infeasible on a graph of this size.

Our experiments indicate that, as far as accuracy goes, our algorithm performs similarly to that of Chepoi-Estillon despite the larger theoretical upper bound on the error. In many cases, in fact, our estimate is better than that one. The p -radius estimated by the algorithm in Theorem 4.1 is always within 1 of their estimate in our trials. Combined with the significant improvement in running time, this makes our algorithm a preferable choice for solving p -centres in practice. For comparison, our implementation of our algorithm terminated in under two seconds on the sn-medium graph, while C-E took about one minute.

While the tree-approximation heuristic is simple, and runs in quasilinear time $O(m + n)$, the approximation guarantee is only as good as the distance approximation of T , hence the additive error could up to $O(\delta \log n)$. However, our experiments show that it seems to perform well in practice and may be a good choice of heuristic in some applications.

³The graphs p2p-gnutella25 and web-stanford are available publicly as part of the Stanford Large Network Dataset Collection. The sn-medium graph is extracted from the social network Facebook, and the sprintlink-1239 graph is an IP-layer network from the Rocketfuel ISP.

Table 2: Comparison of estimates of the p -radius.

	sprintlink-1239			p2p-gnutella25			sn-medium			web-stanford		
	Thm 9	Ch.	Tree	Thm 9	Ch.	Tree	Thm 9	Ch.	Tree	Thm 9	Ch.	Tree
$p = 1$	7	7	8	8	8	7	7	7	8	82		
$p = 2$	7	7	7	8	8	7	7	8	8	59		
	Thm 8	C-E	Tree	Thm 8	C-E	Tree	Thm 8	C-E	Tree	Thm 8	C-E	Tree
$p = 3$	5	6	6	7	7	7	7	7	8	47		
$p = 4$	5	6	6	7	7	7	6	7	8	46		
$p = 5$	4	5	6	7	6	7	6	7	8	44		
$p = 6$	4	5	6	7	6	7	6	6	8	44		
$p = 7$	4	5	5	6	6	7	6	6	8	44		
$p = 8$	4	5	5	6	6	7	6	6	8	38		
$p = 9$	4	5	5	6	6	7	6	6	7	29		
$p = 10$	4	5	5	6	6	7	6	6	7	29		
$p = 11$	4	5	5	6	6	7	6	6	7	27		
$p = 12$	4	4	5	6	6	7	5	6	7	23		
$p = 13$	4	4	5	6	5	7	5	6	7	23		
$p = 14$	4	4	5	6	5	7	5	6	7	23		
$p = 15$	4	4	5	6	5	7	5	6	7	22		
$p = 16$	4	4	5	6	5	7	5	6	7	21		
$p = 17$	4	4	5	6	5	7	5	6	7	20		
$p = 18$	4	4	5	5	5	7	5	6	7	19		
$p = 19$	4	4	5	5	5	7	5	6	7	16		
$p = 20$	4	4	5	5	5	7	5	6	7	17		

References

- [1] Martin R Bridson and André Haefliger. *Metric spaces of non-positive curvature*, volume 319. Springer Science & Business Media, 1999.
- [2] R Chandrasekaran and Andrew Daughety. Location on tree networks: p-centre and n-dispersion problems. *Mathematics of Operations Research*, 6(1):50–57, 1981.
- [3] Victor Chepoi, Feodor Dragan, Bertrand Estellon, Michel Habib, and Yann Vaxès. Diameters, centers, and approximating trees of delta-hyperbolic geodesic spaces and graphs. In *Proceedings of the twenty-fourth annual symposium on Computational geometry*, pages 59–68. ACM, 2008.
- [4] Victor Chepoi and Bertrand Estellon. Packing and covering δ -hyperbolic spaces by balls. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, pages 59–73. Springer, 2007.

- [5] Martin E Dyer and Alan M Frieze. A simple heuristic for the p-centre problem. *Operations Research Letters*, 3(6):285–288, 1985.
- [6] Erhan Erkut. The discrete p-dispersion problem. *European Journal of Operational Research*, 46(1):48–60, 1990.
- [7] Erhan Erkut and Susan Neuman. Comparison of four models for dispersing facilities. *INFOR*, 29:68–86, 1991.
- [8] Erhan Erkut, Yilmaz Ülküsal, and Oktay Yenicerioglu. A comparison of p-dispersion heuristics. *Computers & Operations Research*, 21(10):1103 – 1113, 1994.
- [9] Greg N Frederickson. Parametric search and locating supply centers in trees. In *Algorithms and Data Structures*, pages 299–319. Springer, 1991.
- [10] Mikhael Gromov. *Hyperbolic groups*. Springer, 1987.
- [11] Dorit S Hochbaum and David B Shmoys. A best possible heuristic for the k-center problem. *Mathematics of operations research*, 10(2):180–184, 1985.
- [12] Wen-Lian Hsu and George L Nemhauser. Easy and hard bottleneck location problems. *Discrete Applied Mathematics*, 1(3):209–215, 1979.
- [13] Donald B Johnson. Efficient algorithms for shortest paths in sparse networks. *Journal of the ACM (JACM)*, 24(1):1–13, 1977.
- [14] William Sean Kennedy, Onuttom Narayan, and Iraj Saniee. On the Hyperbolicity of Large-Scale Networks. *ArXiv e-prints*, June 2013.
- [15] SS Ravi, Daniel J Rosenkrantz, and Giri Kumar Tayi. Facility dispersion problems: Heuristics and special cases. In *Algorithms and Data Structures*, pages 355–366. Springer, 1991.
- [16] Douglas R Shier. A min-max theorem for p-center problems on a tree. *Transportation Science*, 11(3):243–252, 1977.